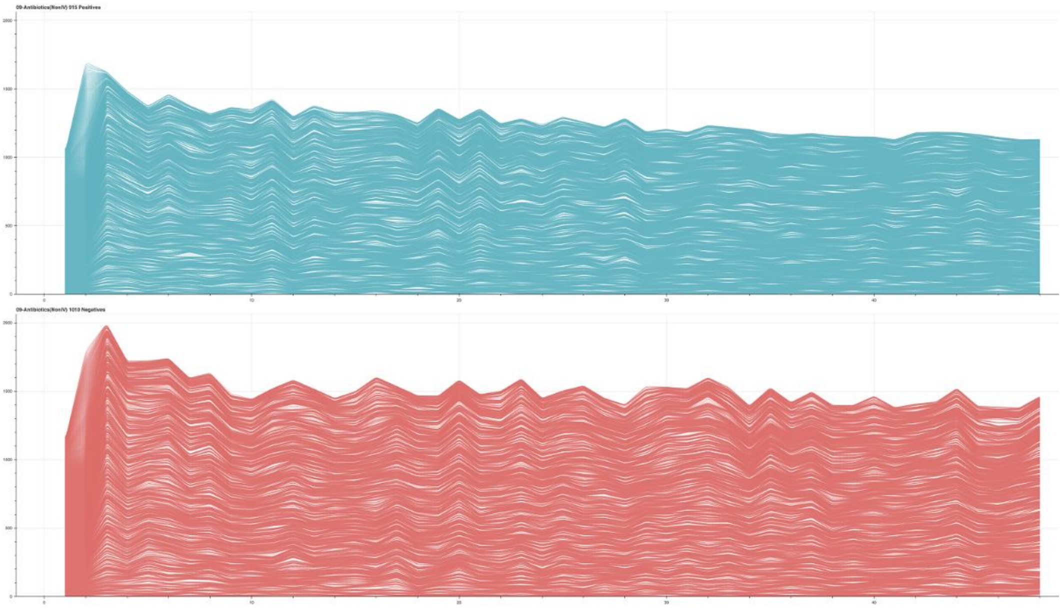
Weekly Report

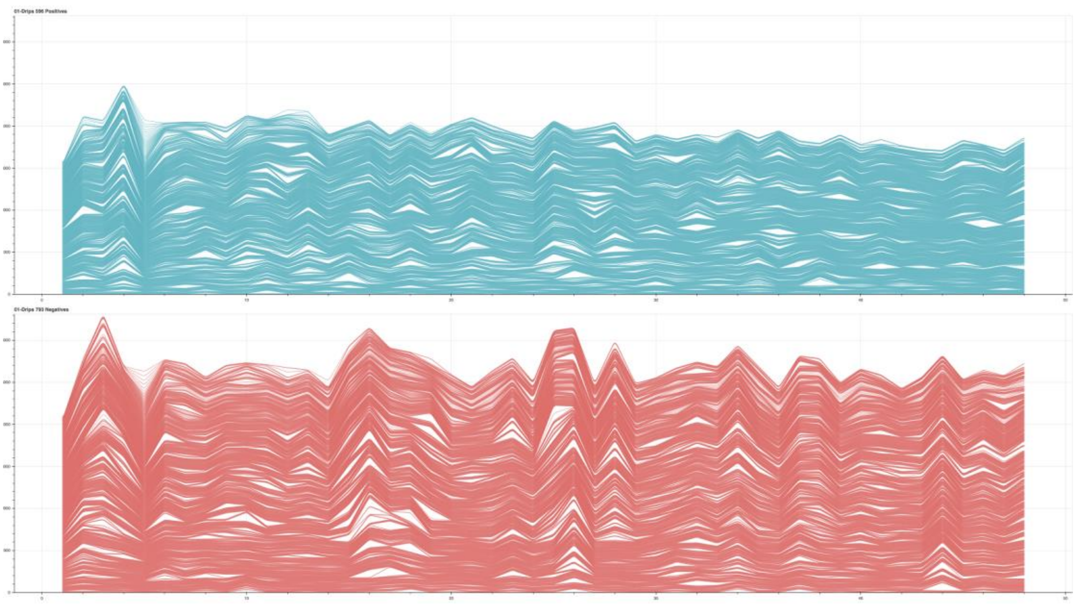
1 Done

1.1 Project with Chuan

We discussed about a new representation, which sort patients by values and pile the lines up. The trend can be found easily. Through textures, we can also detect abnormal patterns.



However, the effects sometimes are not so good. For example, the patient amount could be different. Then, the total height can not be compared directly. So, we need some preprocessing, like sampling. Besides, we sort time series records based on the values on the first timestamp. But the values on the rest timestamps are a mess. We plan to solve this problem by adding interactions. For examples, we can allow users to select a timestamp and sort records through the values on that timestamp.

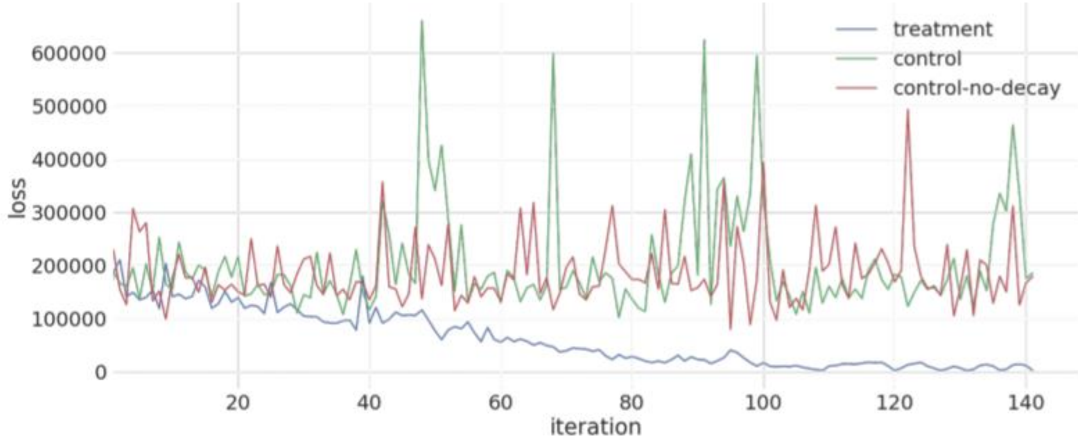


1.2 Paper Reading

I found something to visualize in each iteration from the following paper.

- *Federated Learning*

  This work compare the loss in each iteration.



The author said that “if the model of an iteration is performing worse than the one of the previous iterations, it could be purely because not enough users reported back, which lead to inaccurate estimates”. So, we also need to pay attention to the reported record amount. If the amount is less than a default value, users should decide if there is a need to stop this iteration and avoid result damage.

- *A Performance Evaluation of Federated Learning Algorithms*

  Those two algorithms tell what could happen in an iteration of federated learning.

Algorithm 1: FederatedAveraging

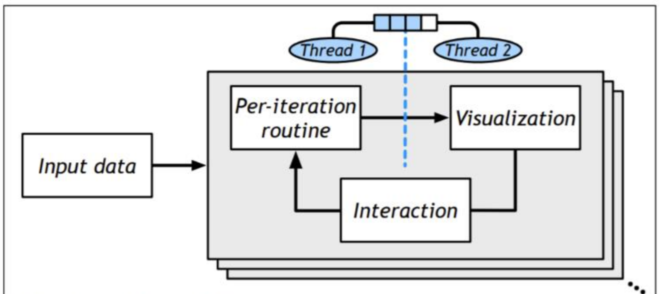
initialize w_0
for *each round* $t = 0, 1, \ldots$ **do**
  $m \leftarrow \max(\lfloor C \cdot K \rfloor, 1)$
  S_t = random set of m clients
  for *each client* $k \in S_t$ ***in parallel*** **do**
    $w_{t+1}^k = \texttt{ClientUpdate}(k, w_t)$
  $w_{t+1} = \sum_{k \in S_t} \frac{n_k}{n_\sigma} w_{t+1}^k, \quad n_\sigma = \sum_{k \in S_t} n_k$

Algorithm 2: Federated SVRG

initialize w_0
$h \leftarrow$ stepsize
$\{\mathcal{P}_k\}_{k=1}^K$ = data partition
for *each round* $t = 0, 1, \ldots$ **do**
  Compute $\nabla f(w_t) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(w_t)$
  for *all K clients* ***in parallel*** **do**
    initialize: $w_{t+1}^k \leftarrow w_t$, and $h_k = \frac{h}{n_k}$
    let $\{i_s\}_{s=1}^{n_k}$ be a permutation of $\mathcal{P}_k$
    for $s = 1, \ldots, n_k$ **do**
      $\Theta \leftarrow \nabla f_{i_s}(w_{t+1}^k) - \nabla f_{i_s}(w_t) + \nabla f(w_t)$
      $w_{t+1}^k \leftarrow w_{t+1}^k - h_k \cdot \Theta$
  $w_{t+1} = \sum_{k=1}^K \frac{n_k}{n} w_{t+1}^k$ `// update global model`

- *PIVE: Per-Iteration Visualization Environment for Real-Time Interactions with Dimension Reduction and Clustering*

  This work shows the results of each iteration.



(b) The PIVE workflow

Also, PIVE allows users to improve model results by interactions. The approach is called hard replacement.

Algorithm 1 Iterative methods

Input: $X = \{x_1, \cdots, x_n\}$ and parameter α
Output: $Y = \{y_1, \cdots, y_n\}$
$t \leftarrow 0$
Initialize $Y^t = \{y_1^t, \cdots, y_n^t\}$
repeat
  $t \leftarrow t + 1$
  /* Per-iteration routine */
  for $i \leftarrow 1, \cdots, n$ **do**
    $y_i^t \leftarrow f(\{X, Y^0, \cdots, Y^{t-1}\}, \alpha)$
  $Y^t \leftarrow \{y_1^t, \cdots, y_n^t\}$
until a stopping criterion is satisfied
$T \leftarrow t$ /* Final iteration index */
$Y \leftarrow Y^T$ /* Final output */

Algorithm 2 Soft and hard replacement interactions

Input: an iteration index t at which a user interaction was performed, interacted data item indices $I = \{i_1, \cdots, i_l\}$, their new values $\{\tilde{y}_{i_1}, \cdots, \tilde{y}_{i_l}\}$, and a new parameter $\tilde{\alpha}$
$\alpha \leftarrow \tilde{\alpha}$
for $i \leftarrow i_1, \cdots, i_l$ **do**
  $y_i^t \leftarrow \tilde{y}_i$
if Hard replacement **then**
  Replace the per-iteration routine in Algorithm 1 as follows:
  /* Per-iteration routine*/
  for $i \leftarrow 1, \cdots, n$ **do**
    if $i \notin I$ **then**
      $y_i^t \leftarrow f(\{X, Y^0, \cdots, Y^{t-1}\}, \alpha)$
    else
      $y_i^t \leftarrow y_i^{t-1}$
Continue from the iteration index $t + 1$ until a stopping criterion is satisfied in Algorithm 1